「パス構築・パス検証」クライアントサンプル実装

Java 版

取扱説明書



2003 年 3 月

情報処理振興事業協会

1 概要

JDK 1.4 以降の JAVA のコアライブラリには、java.security.cert.*に証明書のパス構築・検証のためのクラスが標準的に用意されている。これらのクラスはフレームワークとなっており、エンドユーザは JAVA の規格に従ったコードを作成するだけで、複数のベンダが提供する実装を選択的に使用することが出来る。

本書はこのクラス群を使用して、サンプルクライアントと証明書のパス構築・検証を行うプロバイダを開発したプログラム群の取扱いについて記述する。

- 署名生成プログラム
  - gpkisign
- パス構築・検証プログラム
  - gpkiverify（オリジナルの実装を利用したサンプルプログラム）
  - pkixverify（SUN の実装を利用したサンプルプログラム）

2 実行環境

JAVA サンプル実装は以下の条件を満たす環境にて稼動する。

- JAVA JDK 1.4 相当以上。
- 2003 年 2 月 21 以降の富士ゼロックス提供のプロバイダがインストールされている。

3 サンプル実装のインストール

JAVA サンプル実装は表 1 のようなファイルにより構成される。

表 1　配布ファイルの一覧

	ファイル名	役割
1	readme.txt	リリースノート
2	gpkisign	署名生成プログラム
3	gpkiverify	パス構築・検証プログラム（オリジナルプロバイダ使用版）
4	pkixverify	パス構築・検証プログラム（SUN 実装プロバイダ使用版）
5	GPKI.jar	オリジナルプロバイダとサンプルプログラムを含むアーカイブ
6	fujixerox-security.jar	富士ゼロックス提供のアーカイブ
7	fujixerox-PKI.jar	富士ゼロックス提供のアーカイブ

JAVA サンプル実装のインストール手順は以下の通りである。

1. gpkiverify/pkixverify/gpkisign を適切な箇所にコピーする。

2. GPKI.jar/fujixerox-security.jar/fujixerox.-PKI.jar は
、${JAVA_HOME}/jre/lib/ext/以下にコピーする。
3. ${JAVA_HOME}/jre/lib/security/java.security を編集し、図 1 太文字で示すエントリィを追加する。

図１ プロバイダのインストール

```
#
# List of providers and their preference orders (see above):
#
security.provider.1=sun.security.provider.Sun
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
security.provider.3=com.sun.rsajca.Provider
security.provider.4=com.sun.crypto.provider.SunJCE
security.provider.5=sun.security.jgss.SunProvider
security.provider.6=org.jnsa.ChallengePKI2002.Provider
security.provider.7=jp.co.fujixerox.security.Provider
```

4 プログラムの実行

4.1 署名テストプログラム

4.1.1 機能説明

gpkisign は、PKCS#7 を用いてデジタル署名処理を実行するためのプログラムである。

指定された秘密鍵及びそれに関連する証明書を利用して、対象データに署名処理を実行する。対象データが指定されていない場合は、空データをテスト用にセットし署名処理を行う。

署名処理結果は、標準出力に出力される。

成功の場合は特にメッセージを出力せず、終了する。失敗の場合は失敗理由を出力する。

署名されたデータは、署名データとして、`元のファイル名.p7m'として保存される。

4.1.2 コマンドラインの書式

gpkisign [オプション] cert<1> cert<2 >...cert<n>

4.1.3 コマンドラインオプション

以下の引数を指定する。

cert<1> cert<2 >...cert<n>

　署名処理に使用する秘密鍵に関連する証明書を任意数指定可能。複数指定する場合は、各引数間を空白（スペース）で区切る。-c により任意のクレデンシャルが指定されている場合は、何も指定しなくてもよい。ここで指定する証明書のデータ形式は、DER 形式とする。

表 2 gpkisign コマンドオプション一覧

オプション	オプションの引数	必須項目	説明
-v	なし	ー	このテストプログラムが基づいているテストプログラム仕様のバージョン番号を表示する。
-k	File 名	○	署名処理に使用する秘密鍵を指定する。クレデンシャルファイルは、PKCS#8 とする。
-d	File 名	○	署名処理を行う対象となるデータファイルを指定する。ファイルに格納されているデータ形式は問わない。
-f	Path 名		署名処理を行う対象となるデータファイルが格納されているフォルダ/ディレクトリを指定する。このオプションによりデータが指定される場合は、path で指定されたフォルダ/ディレクトリ以下全てのファイルが署名対象となる。パス内のファイルに格納されているデータ形式は問わない。
-o	File 名	ー	署名処理済み（署名結果）ファイル名を指定する。省略可能。省略時のデフォルト値は、`元のデータファイル名.p7m’。
-p	Passphrase	○	-k で指定された署名を行うために使用する秘密鍵に対するパスフレーズを入力する。

○：指定必須　ー：指定任意。

4.1.4 使用例

秘密鍵 key.p8 を使用して、ファイル test.txt に署名処理を行う。

```
gpksign –k key.p8 –d test.txt
```

4.1.5 終了ステータス

以下の終了ステータスが返される。

0	正常終了
<0	内部エラー（システムエラー）が発生した。

4.2 署名検証テストプログラム

4.2.1 機能説明

gpkiverify/pkixverify は署名検証テストプログラムである。

gpkiverify はオリジナルの実装を利用したサンプルプログラムであり、pkixverify は SUN の実装を利用したサンプルプログラムである。

gpkiverify/pkixverify は指定された証明書及びポリシーを利用して、対象データの署名検証を行う。上記引数の他に、任意の設定ファイル等による初期値指定方法を備えなければならない。

署名検証結果は標準出力に出力される。失敗の場合は失敗理由も併せて出力する。

4.2.2 コマンドラインの書式

gpkiverify/pkixverify [オプション] ipcy<1> ipcy<2 > … ipcy<n>

4.2.3 コマンドラインオプション

以下の引数が指定できる。

ipcy<1> ipcy<2 > … ipcy<n>

initial-policy-set を任意数指定することができる。複数指定する場合は、各引数間を空白（スペース）で区切る。省略可能。省略時は、NULL を示す。規定値として、以下の値が使用可能である。

“ANY”　：全てのポリシーを受け入れる場合(any-policy)。

表 3 gpkiverify/pkixverify コマンドオプション一覧

オプション	オプションの引数	必須項目	説明
-v	なし	—	このテストプログラムが基づいているテストプログラム仕様のバージョン番号を表示する。実装必須。
-f	Path 名	○	コマンドラインに指定されているファイルが格納されているフォルダ/ディレクトリを指定する。
-c	File 名	△	検証対象証明書ファイル（der 形式）を指定する。
-t	File 名	○	署名･証明書検証側トラストアンカーの自己署名証明書ファイル（der 形式）を指定する。
-P	[File 名]	—	中間 CA の証明書ファイルを指定する。ファイルを複数指定する場合は、コロン「:」で区切る。※
-ip	なし	—	initial-policy-mapping-inhibit フラグの有効/無効を指定する。-ip オプションがある場合、有効となる。
-ie	なし	—	initial-explicit-policy フラグの有効/無効を指定する。-ie オプションがある場合、有効となる。
-ia	なし	—	initial-any-policy-inhibit フラグの有効/無効を指定する。-ia オプションがある場合、有効となる。
-l	Host 名	—	LDAP サーバのホストとポート番号を「IP アドレス:ポート番号」もしくは「FQDN:ポート番号」の形式で指定する。

○：実装必須 △：いずれかが必須 —：指定装任意。

※各 CA が発行する CRL/ARL については、各 CA 証明書（トラストアンカー、中間 CA 証明書）のファイル名の拡張子が crl あるいは arl となっているものを読み込む。これらのファイルは全て-f で指定されたパス配下にあるものとする。

4.2.4 使用例

【例 1】

`/project/date` 配下にある `target.1.1.1.crt` を検証対象証明書ファイルとし、`trust.1.1.1.crt` をトラストアンカー証明書ファイルとして、証明書リポジトリとして `ldap.gpki.go.jp` を使用し、証明書検証をオンラインモードで行う。

```
gpkiverify –f /project/data –c target.1.1.1.crt –t trust.1.1.1.crt –l ldap.gpki.go.jp:389
```

出力

```
Target is verified
```

【例 2】

`/project/date` 配下にある `target.1.1.1crt` を検証対象証明書ファイルとし、`trust.1.1.1.crt` をトラストアンカー証明書ファイルとして、証明書検証のみをオフラインモードで行う。

`intermediate.1.1.1.crt` 及び `intermediate.1.1.2.crt` を中間 CA の証明書ファイルとし、トラストアンカーや中間 CA の CRL/ARL については、それぞれのファイル名の拡張子が `crl` または `arl` となっているファイルを参照する。

```
gpkiverify –f /project/data –c target.1.1.1.crt –t trust.1.1.1.crt –P intermediate.1.1.1.crt:intermediate.1.1.2.crt
```

出力

```
Can not verify target certificate
```

4.2.5 終了ステータス

以下の終了ステータスが返される。

0	正常終了
>0	検証エラーが発生した。